2-3 Tree Insertion

Michael Perlman

CS 323 – D&A of Algorithms

Fall 2013

Dr. Phillips

2-3 Tree insertion algorithm steps:

1. Prepare files for input/output
2. spot <- findSpot(t, data)
3. Insert new data under spot
   1. Case 0: spot has no children  
      spot.child1 <- data
   2. Case 1: spot has one child  
      Sort child1 and new data in ascending order, and insert them under child1 and child2, respectively
   3. Case 2: spot has two children  
      Sort child1, child 2, and new data in ascending order, and insert them under child1, child2, and child3 respectively
   4. Case 3: Spot has 3 children
      1. Arrange all 4 items (3 children + 1 new data) in ascending order
      2. Split 4 nodes into two groups of 2
      3. Create new internal node, new, at same level as spot
      4. Let spot take the first group of leaf nodes, and new take second group
      5. Update keys of spot and new
      6. Insert new under spot’s parent

findSpot algorithm steps:

Node\* findSpot(Node\* T, int elem){

1. If T.child1 if a leaf node, return T
2. If T.key1 > elem, T=T.child1  
   If elem > T.key1 and elem <= T.key2, T=T.child2  
   If elem > T.key2 and T.key2 == -1, T=T.child3
3. Repeat steps 1-2 until T.child1 is a leaf node.

Pre-order traversal printing algorithm:

void preOrderTraversal(TreeNode T){

if(T == NULL) return;

else {

print(T);

preOrderTraversal(T.child1);

preOrderTraversal(T.child2);

preOrderTraversal(T.child3);

}

}